Міністерство освіти і науки України

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

Звіт

з лабораторної роботи №9 з дисципліни

«Програмування - 2. Структури даних та алгоритми»

**«ДОСЛІДЖЕННЯ ВКАЗІВНИКІВ НА ФУНКЦІЇ У МОВІ С++ ТА ДЕЛЕГАТІВ С#»**

Виконав студент

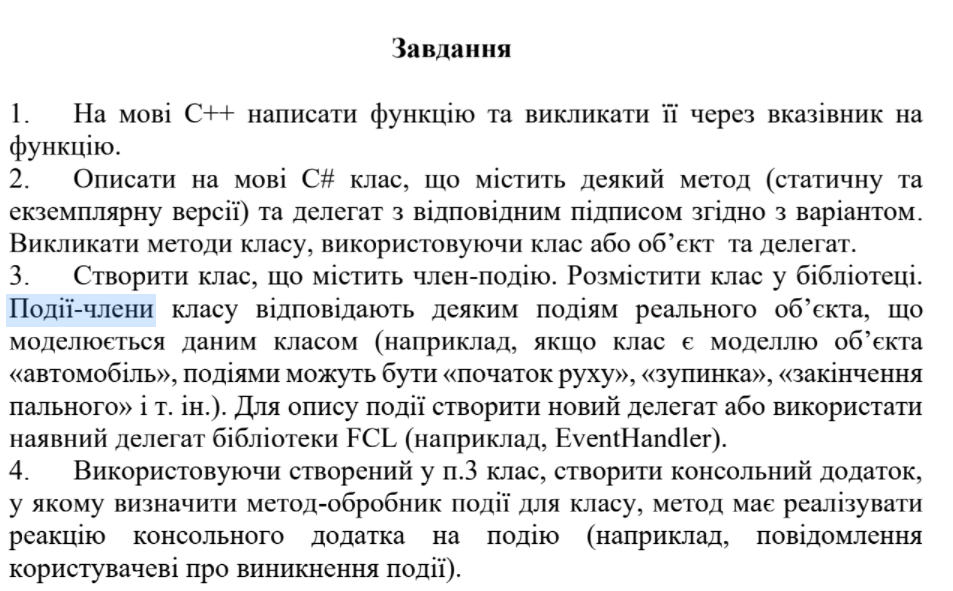
**ІС-01 Кубай Дмитро**

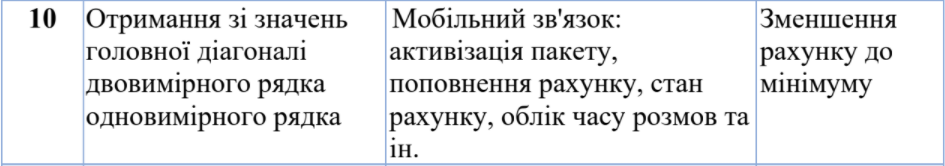
Перевірив викладач

**Проскура С. Л.**

**Київ – 2021**

1. **Завдання:**





1. **Код програми:**

**C++**

**Source.cpp**

#include <iostream>

using namespace std;

int\* GetArray(int \_elements[2][2]);

int main()

{

int\* (\*operation)(int arr[2][2]);

operation = GetArray;

int arr[2][2] = { {1, 5, },

{1, 2 } };

int\* new\_arr = operation(arr);

}

int\* GetArray(int \_elements[2][2])

{

int arr[2] = { \_elements[0][0], \_elements[1][1] };

cout << "Diagonal array: " << arr[0] << ", " << arr[1] << "\n";

return arr;

}

**C#**

**Program.cs**

using System;

using Lab09Library;

namespace Lab09sharp

{

class Program

{

static void Main(string[] args)

{

int[,] arr = new int[,] { { 1, 5 },

{ 1, 2 }

};

Arr new\_arr = new Arr(arr, DisplayMessage);

int[] line = new\_arr.GetArray();

Console.WriteLine("\n/////////////////////////////\n");

Phone nokia = new Phone("nokia", DisplayPhoneMessage);

nokia.ActivatePhone();

nokia.Put(500);

nokia.Talk(2);

nokia.CheckStatus();

}

private static void DisplayMessage(string message)

{

Console.WriteLine(message);

}

private static void DisplayPhoneMessage(object sender, PhoneEventArgs e)

{

Console.WriteLine(e.Message);

}

}

}

**Phone.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace Lab09Library

{

public class Phone

{

private event PhoneStateHandler Notify;

private int \_talk\_time = 0;

private decimal \_sum = 0;

private bool \_activated = false;

private string \_name = null;

public Phone(string name, PhoneStateHandler e)

{

\_name = name;

this.Notify += e;

Notify?.Invoke(this, new PhoneEventArgs("Phone " + \_name + " was created", \_sum));

}

public void ActivatePhone()

{

\_activated = true;

Notify?.Invoke(this, new PhoneEventArgs("Your phone " + \_name + " was activated", \_sum));

}

public void Put(decimal money)

{

\_sum += money;

Notify?.Invoke(this, new PhoneEventArgs("Your balance " + \_sum , \_sum));

}

public void Talk(int hour)

{

if (\_activated)

{

if (hour > 0)

{

\_talk\_time += hour;

\_sum -= hour \* 10;

Notify?.Invoke(this, new PhoneEventArgs("You has talked for all time: " + \_talk\_time, \_sum));

}

}

else

{

Notify?.Invoke(this, new PhoneEventArgs("Your phone wasnt activated ", \_sum));

}

}

public void CheckStatus()

{

if(\_activated)

Notify?.Invoke(this, new PhoneEventArgs("Your phone is activated and your current balance is: "

+ \_sum +",\nyou has talked " + \_talk\_time + " hours for all time", \_sum));

else

Notify?.Invoke(this, new PhoneEventArgs("Your phone wasnt activated ", \_sum));

}

}

}

**Arr.cs**

using System;

namespace Lab09Library

{

public class Arr

{

public delegate void ArrHandler(string message);

private event ArrHandler Notify;

private int[,] \_elements;

public Arr(int[,] arr, ArrHandler ev)

{

\_elements = arr;

Notify += ev;

}

public int[] GetArray()

{

int[] arr = new int[] {\_elements[0, 0], \_elements[1, 1]};

Notify?.Invoke("Your array: " + arr[0] + ", " + arr[1]);

return arr;

}

}

}

**PhoneStateHandler.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace Lab09Library

{

public delegate void PhoneStateHandler(object sender, PhoneEventArgs e);

public class PhoneEventArgs

{

public string Message { get; private set; }

public decimal Sum { get; private set; }

public PhoneEventArgs(string mes, decimal sum)

{

Message = mes;

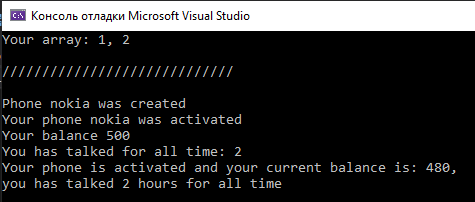
Sum = sum;

}

}

}

1. **Виконання програми**

****

1. **Висновок**

У даній роботі було дослідити механізм обробки подій у мові С#, навчитися

створювати та використовувати делегати і написати програму, яка реалізує:

клас, що містить деякий метод (статичну та екземплярну версії) та делегат з відповідним підписом згідно з варіантом. Викликати методи класу, використовуючи клас або об’єкт та делегат. Потрібно було створити клас, що містить член-подію, розмістити клас у бібліотеці. Події-члени класу відповідають деяким подіям реального об’єкта, що моделюється даним класом (наприклад, якщо клас є моделлю об’єкта «автомобіль», подіями можуть бути «початок руху», «зупинка», «закінчення пального» і т. ін.). Для опису події створити новий делегат або використати наявний делегат бібліотеки FCL (наприклад, EventHandler).

Я, використовуючи створений у п.3 клас, створив консольний додаток,

у якому визначив метод-обробник події для класу, метод реалізує реакцію консольного додатка на подію